Amendment to the Specification

Please replace the paragraph beginning on page 1, line 10, with the rewritten paragraph below.

Modern microprocessors offer unprecedented performance. For a variety of digital integrated circuits (IC's), speed, level of integration (i.e. transistors per square centimeter) and capabilities have improved. Moreover, in many cases, these performance improvements have been accompanied by reductions in size, power consumption and cost of the devices. However, these benefits ~~have~~ required greater complexity in digital logic design. Because of this complexity, the investment of time and resources by the manufacturer to design and fabricate a digital logic device has increased. For this same reason, the possibility of a mistake or oversight on the part of the designer has become more likely.

Please replace the paragraph beginning on page 5, line 18, with the rewritten paragraph below.

Fig. 4 is a block diagram of logic used by a prior art hardware break module for handling fix-up cycles; and.

Please replace the paragraph beginning on page 15, line 1, with the rewritten paragraph below.

Since this instruction sequence includes a branch instruction, the behavior of the pipeline depends on whether or not the branch to "go_here" is taken. In this example, it is assumed that the branch will not be taken, so all of the instructions are executed in the order in which they appear in memory. Note that the third instruction (LW) occurs in a branch delay slot, and that a data load miss occurs ~~in~~ during processor clock cycle T7, in the WB pipeline stage of this instruction. A fix-up cycle is inserted to allow the data required by the LW instruction to be fetched from memory. Processing of all of the pending instructions in the pipeline is temporarily suspended during the fix-up cycle. Thus, for example, the fourth instruction does not advance beyond the MEM pipeline stage from T7 to T8. On the other hand, the branch instruction has already left the pipeline by the time the fix-up cycle occurs – this event is recorded by the complex break logic within the state machine (item 24 in Fig. 2), as described in greater detail below.